AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions and listings of claims in the application:

1. (Currently Amended) A computer-readable medium encoded with a computer program ~~product, tangibly embodied on a machine-readable storage device,~~ comprising instructions ~~operable to cause data processing apparatus to~~ that, when executed, operate to cause a computer to perform operations comprising:

~~establish~~ establishing a plurality of checkpoints in a computer program, the computer program having a program structure, each checkpoint in the plurality of checkpoints including an assertion statement;

~~assign~~ assigning each checkpoint in the plurality of checkpoints to a checkpoint group without regard to the program structure of the computer program, the assignment of each checkpoint to a checkpoint group being specified in the statement defining the respective checkpoint; ~~and~~

~~associate~~ associating each checkpoint group with one of a plurality of activation variants that indicates a behavior based on a result of the assertion statement, wherein checkpoint groups associated with an activation variant behave in accordance with the activation variant;

executing a non-activatable checkpoint; and

selectively executing at least one checkpoint of the plurality of checkpoints based on an activation status of the checkpoint group.

2. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, wherein the checkpoints comprise breakpoint statements.

3. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, wherein the operations further comprise ~~comprising instructions to~~:

~~establish~~ establishing activation variants to enable multiple checkpoint groups to be managed jointly.

4. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, wherein the operations further comprise ~~comprising instructions to~~:

~~receive~~ receiving a control input activating a first checkpoint group; and

~~activate~~ activating the checkpoints in the first checkpoint group.

5. (Currently Amended) The ~~product~~ computer-readable medium of claim 4, wherein the operations ~~instructions to receive a control input~~ further comprise ~~instructions to~~:

~~receive~~ receiving a control input that specifies a mode in which checkpoints that are assertions terminate on assertion failure;

~~receive~~ receiving a control input that specifies a mode in which checkpoints that are assertions log status on assertion failure; and

~~receive~~ receiving a control input that specifies a mode of activating checkpoints in which assertions break in a debugger on assertion failure.

6. (Currently Amended) The ~~product~~ computer-readable medium of claim 4, wherein the operations further comprise ~~comprising instructions to~~:

~~receive~~ receiving a control input specifying that activating is to be performed only for a particular user of multiple users using the computer program, the activating not affecting the use of the computer program by other users.

7. (Currently Amended) The ~~product~~ computer-readable medium of claim 4, wherein the operations further comprise ~~comprising instructions to~~:

~~receive~~ receiving a control input specifying that activating is to be performed only for a particular server of multiple servers on which the computer program is running.

8. (Cancelled)

9. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, wherein:

each assertion statement when activated testing whether a specified assertion condition is true or false; and

the checkpoints comprise breakpoint statements, each breakpoint statement when activated halting program execution when it is encountered during program execution.

10. (Currently Amended) The ~~product~~ computer-readable medium of claim 2, wherein:

the assertion statements comprise an assertion statement having an argument to activate logging with programmer-controlled granularity, the argument being used to determine whether to update a log entry when the assertion statement fails.

11. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, further comprising instructions to establish a development environment for developing the computer program in which the checkpoint groups are development objects.

12. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, wherein the checkpoints and the computer program are in a compiled form.

13. (Currently Amended) ~~Apparatus~~ An apparatus, comprising:

means for establishing a plurality of checkpoints in a computer program, the computer program having a program structure, each checkpoint in the plurality of checkpoints including an assertion statement;

means for assigning each checkpoint in the plurality of checkpoints to a checkpoint group without regard to the program structure of the computer program, the assignment of each checkpoint to a checkpoint group being specified in the statement defining the respective checkpoint; ~~and~~

means for associating each checkpoint group with one of a plurality of activation variants that indicates a behavior based on a result of the assertion statement, wherein checkpoint groups associated with an activation variant behave in accordance with the activation variant;

means for executing a non-activatable checkpoint, and for selectively executing at least one checkpoint of the plurality of checkpoints based on an activation status of the checkpoint group.

14. (Previously Presented) The apparatus of claim 13, wherein:
the checkpoints comprise breakpoints.

15. (Cancelled)

16. (Previously Presented) The apparatus of claim 13, further comprising:
means for associating an activation variant with a compilation unit.

17. (Currently Amended) A method, comprising:

receiving a computer program having a plurality of checkpoints, each checkpoint being assigned to at least one of a plurality of checkpoint groups, each checkpoint and each checkpoint group being identified by a group identifier, each checkpoint in the plurality of checkpoints including an assertion statement, the assignment of each checkpoint to a checkpoint group being specified in the statement defining the respective checkpoint, the statement including the group identifier identifying the checkpoint group;

associating each checkpoint group with one of a plurality of activation variants that indicates a behavior based on a result of the assertion statement, wherein checkpoint groups associated with an activation variant behave in accordance with the activation variant; ~~and~~

receiving user input to invoke checkpoints as a group according to their group identifiers;

executing a non-activatable checkpoint; and

selectively executing at least one checkpoint of the plurality of checkpoints based on an activation status of the checkpoint group.

18. (Previously Presented) The method of claim 17, further comprising:

receiving a user input specifying a mode of invocation of checkpoints; and

invoking checkpoints according to the specified mode.

19. (Previously Presented) The method of claim 17, further comprising:

receiving a further user input specifying a scope of invocation of checkpoints, the scope specifying that checkpoints are to be invoked only for a particular user of multiple users using the computer program; and

invoking checkpoints according to the specified scope.

20. (Previously Presented) The method of claim 17, further comprising:

receiving a further user input specifying a scope of invocation of checkpoints, the scope specifying that checkpoints are to be invoked only for a particular server of multiple servers on which the computer program is running; and

invoking checkpoints according to the specified scope.

21.-23. (Cancelled)

24. (Previously Presented) The method of claim 17, wherein the computer program has checkpoints including breakpoints.

25. (Currently Amended) A method for adding checkpoints to a computer program having source code, the method comprising:

adding to the computer program a plurality of checkpoints each assigned to a checkpoint group by a respective group name for the checkpoint, each checkpoint in the plurality of checkpoints including an assertion statement, the assignment of each checkpoint to a checkpoint group being specified in the statement defining the respective checkpoint; ~~and~~

associating each checkpoint group with one of a plurality of activation variants that indicates a behavior based on a result of the assertion statement, wherein checkpoint groups associated with an activation variant behave in accordance with the activation variant; and adding to the computer program at least one checkpoint as a non-activatable checkpoint, which non-activatable checkpoint is executed regardless of an activation status of the checkpoint group.

26. (Previously Presented) The method of claim 25, further comprising:

adding the plurality of checkpoints to the source code of the computer program, the respective group name for each checkpoint being included in the source code for the checkpoint; and

transporting the checkpoint groups as development objects with the computer program from a development environment to a production environment, the development objects being objects created and managed by the development environment.

27. (Currently Amended) The ~~product~~ computer-readable medium of claim 10, wherein:

the argument to activate logging indicates that a log entry is made for each distinct value of a named field.

28. (Currently Amended) The ~~product~~ computer-readable medium of claim 1, wherein the checkpoint groups and the activation variants are established in a maintenance module, and affect operation of a separate debugger module.

29. (Previously Presented) The apparatus of claim 13, wherein the means for establishing, the means for assigning, and the means for associating are provided in a maintenance module, and wherein the checkpoint groups and the activation variants affect operation of a separate debugger module.

30. (Previously Presented) The method of 17, wherein the checkpoint groups and the activation variants are established in a maintenance module, and affect operation of a separate debugger module.

31. (Previously Presented) The method of 25, wherein the checkpoint groups and the activation variants are established in a maintenance module, and affect operation of a separate debugger module.